AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A system for <u>customizing a core product</u> ~~at least one of specializing, replacing, and adding services of a service oriented architecture~~, the system comprising:

a core product comprising core code, the core product being <u>a service-oriented architecture</u> for use by more than one customer;

at least one hardware server, the at least one hardware server providing a framework for customizing the core product to create a customized <u>service-oriented architecture</u> ~~core product~~, the customized <u>service-oriented architecture</u> ~~core product~~ meeting a service need of at least one specific customer that is not met by the core product;

wherein the ~~core product~~ <u>service-oriented architecture</u> comprises a plurality of existing service implementations;

wherein the at least one hardware server utilizes the framework to create the customized ~~core product~~ <u>service-oriented architecture</u> by at least one of:

specializing at least one of the plurality of existing service implementations; <u>and</u>

replacing at least one of the plurality of existing service implementations; ~~and~~

~~adding a new service implementation;~~

wherein the customized <u>service-oriented architecture</u> ~~core produc~~t includes and functionally utilizes the core product in unaltered form along with any specialized~~,~~ <u>or</u> replacing~~, or added~~ service implementations; and

wherein the at least one of specializing~~,~~ <u>and</u> replacing~~, and adding~~ comprises generation of new code based on the service need of the at least one specific customer.

2. (Previously Presented) The system of claim 1, wherein the framework comprises:

wherein the plurality of existing service implementations are defined in an XML configuration;

a service client for requesting a service implementation;

a service factory for creating the requested service implementation; and

3

a service interface for allowing access to the requested service implementation by the service client.

3. (Previously Presented) The system of claim 2, wherein, if the at least one existing service implementation is specialized, a new custom service implementation is created and the at least one existing service implementation is subclassed.

4. (Previously Presented) The system of claim 3, wherein select methods of the at least one existing service implementation are overridden by the new custom service implementation.

5. (Previously Presented) The system of claim 2, wherein, if the at least one existing service implementation is replaced, a new custom service implementation is created and the at least one existing service implementation is replaced with the new custom service implementation.

6. (Original) The system of claim 2, wherein, if a new custom service is added, a new custom service implementation, a new custom service factory, a new custom service client, and a new custom service interface are created.

7. (Previously Presented) The system of claim 2 , further comprising at least one middleware for accessing a particular service, wherein the service client remains independent of the at least one middleware.

8. (Original) The system of claim 7, wherein the at least one middleware comprises at least one of Web Services, EJB local access, EJB remote access, local Java call access, and MDB message queue access.

9. (Previously Presented) The system of claim 7, wherein a plurality of middleware bindings for the at least one middleware are automatically generated during a build operation.

4

10. (Previously Presented) The system of claim 9, wherein the plurality of middleware bindings are generated via templates.

11. (Original) The system of claim 10, wherein a middleware binding for a new middleware is generated automatically via a new template.

12. (Currently Amended) A method for <u>customizing a core product</u> ~~at least one of specializing, replacing, and adding services of a service oriented architecture~~, the method comprising the steps of:

creating a core product for utilization by more than one customer, the core product comprising core code<u>, the core product being a service-oriented architecture</u>;

on at least one hardware server, creating a framework for customizing the core product to create a customized <u>service-oriented architecture</u> ~~core product~~, the customized <u>service-oriented architecture</u> ~~core product~~ meeting a service need of at least one specific customer that is not met by the core product;

wherein the <u>service-oriented architecture</u> ~~core product~~ comprises a plurality of existing service implementations;

wherein the framework creates the customized <u>service-oriented architecture</u> ~~core product~~ by at least one of:

specializing at least one of the plurality of existing service implementations; <u>and</u>

replacing at least one of the plurality of existing service implementations; ~~and~~

~~adding a new service implementation;~~

wherein the customized <u>service-oriented architecture</u> ~~core product~~ includes and functionally utilizes the core product in unaltered form along with any specialized~~,~~ <u>or</u> replacing~~,~~ ~~or added~~ services; and

wherein the at least one of specializing~~,~~ <u>or</u> replacing~~, and adding~~ comprises generation of new code based on the service need of the at least one specific customer.

13. (Previously Presented) The method of claim 12, wherein the step of creating a framework comprises the steps of:

creating a first service implementation as defined in an XML configuration;

requesting, by a service client, a service implementation;

creating, by a service factory, the first service implementation; and

allowing access, by a service interface, to the first service implementation by the service client.

14. (Previously Presented) The method of claim 13, further comprising the step of specializing the first service implementation by creating a new custom service implementation and subclassing the first service implementation.

15. (Original) The method of claim 14, further comprising the step of overriding select methods of the first service implementation by the new custom service implementation.

16. (Original) The method of claim 13, further comprising the step of replacing the first service implementation by creating a new custom service implementation and replacing the first service implementation with the new custom service implementation.

17. (Original) The method of claim 13, further comprising adding a new custom service implementation by creating a new custom service implementation, a new custom service factory, a new custom service client, and a new custom service interface.

18. (Previously Presented) The method of claim 13 , further comprising the step of accessing a particular service via at least one middleware, wherein the service client remains independent of the at least one middleware.

19. (Original) The method of claim 18, wherein the at least one middleware comprises at least one of Web Services, EJB local access, EJB remote access, local Java call access, and MDB message queue access.

6

20. (Original) The method of claim 18, further comprising automatically generating middleware bindings for the at least one middleware during a build operation.

21. (Original) The method of claim 20, wherein the step of automatically generating comprises generating the middleware bindings via templates.

22. (Original) The method of claim 21, further comprising automatically generating a middleware binding for a new middleware via a new template.

23. (Previously Presented) The system of claim 7, wherein a client proxy stub is instantiated to allow access to the particular service through the at least one middleware.

24. (Previously Presented) The method of claim 18, wherein accessing the particular service via the at least one middleware comprises utilizing a client proxy stub.

25. (Previously Presented) The system of claim 1, wherein the framework creates a customized core product by specializing at least one of the plurality of existing service implementations.

26. (Previously Presented) The system of claim 1, wherein the framework creates a customized core product by replacing at least one of the plurality of existing service implementations.